

**APPLICATION
FOR
UNITED STATES LETTERS PATENT**

TITLE: USB DEVICE

APPLICANT: Nicolas DRABCZUK

22511
PATENT TRADEMARK OFFICE

"EXPRESS MAIL" Mailing Label Number: **EV535680174US**

Date of Deposit: **January 12, 2004**

USB DEVICE

BACKGROUND OF INVENTION

Field of the Invention

[0001] The invention relates to a system comprising a main device arranged to communicate with an auxiliary device. The main device can be, for example, a USB host. The auxiliary device can be, for example, a USB device, which communicates with the USB host via a USB bus using the USB protocol.

Background Art

[0002] The system comprises a USB host, which is connected to various USB devices via a USB bus. The USB host communicates with the USB devices with the USB protocol. The USB protocol allows connecting several USB devices on the same USB bus using a system of time-sharing based on addressed devices. The USB protocol is organized as a master/slave architecture, the USB host is thus responsible of the time-sharing management.

[0003] The USB host may comprise various applications. One or several services may be needed to run an application. An application uses one or several drivers to access and use the associated services. The drivers may be on the USB host.

[0004] The USB device may comprise various services, in particular those, which are needed by the applications of the USB host. A service may be offered, for example, at the device level (standard USB device), or at the interface level (composite USB device).

[0005] The USB device is organized into several levels a device level, a configuration level, an interface level and an endpoint level. Each level is represented by different USB descriptors.

[0006] - A device descriptor describing the overall device. The device descriptor may be associated to one or more configuration descriptors.

- [0007] - A configuration descriptor describing the electrical characteristics of the USB device, or of a part of the USB device. The configuration descriptor may be associated to one or more interface descriptors.
- [0008] - An interface descriptor describing a particular service of the USB device. An interface may contain one or more alternate settings. The interface descriptor may be associated to zero or more endpoint descriptors.
- [0009] - An endpoint descriptor describing a communication channel used by the service defined by the interface descriptor.
- [0010] In a plugging step, the USB device is plugged onto a USB port of the USB host.
- [0011] In an enumeration step, all the USB descriptors are then retrieved from the USB device to the host device. The enumeration step is triggered off with the modification of the voltage level on the line D+ or D- (depending on the USB device speed) due to a pull-up resistor present in the USB device on one of the lines.
- [0012] In a loading step, the USB Host then uses the descriptors to load all the drivers of the USB device. The number of drivers loaded depends on the number of different services present in the USB device. For example, if the USB device is at the same time a scanner and a printer the USB device will have to present two interfaces during the enumeration step. In that case, two drivers, one associated to the scanner interface, and the other one associated to the printer interface, will be loaded. A main driver associated to the device itself could also be loaded.

SUMMARY OF INVENTION

- [0013] An object of the invention is to reduce the costs and to enhance the security.
- [0014] According to one aspect of the invention, a system comprising a main device and an auxiliary device arranged to co-operate with each other, the auxiliary device being arranged to effect a core functionality, the auxiliary device comprising descriptors, characterised in that at least one descriptor defines a functionality that is different from the core functionality.

[0015] The auxiliary device can be for, example, a USB device in the form of a smart card. The main device can be, for example, a USB host, in particular a computer. The smart card may comprise as a core functionality, for example, an internet login application. The functionality that is different from the core functionality can be, for example, a mass storage functionality.

[0016] Thanks to the invention, the Smart Card (USB device) is seen as a mass storage. It is then possible to install a driver and/or an application from the USB device by simulating that the USB device is a mass storage. The USB device can thus be used in any USB host, even if the driver(s) are not installed nor available, since the driver(s) is(are) available in the USB device itself. A user can thus use, for example, the login application directly from the USB device.

[0017] In addition, an Internet Service Provider can, for example, define its own proprietary login application and store it on the Smart Card itself (USB device). The risk of hacking the login application is therefore reduced.

[0018] The invention thus enables an enhanced security and a reduction of the cost.

BRIEF DESCRIPTION OF DRAWINGS

[0019] Fig. 1 illustrates a system comprising a USB host and a USB device;

[0020] Fig.2 illustrates the structure of the USB device;

[0021] Fig. 3 illustrates a method of using the system;

[0022] Fig. 4 illustrates a system comprising a USB host and a USB device;

[0023] Fig. 5 illustrates a method of using the system; and

[0024] Fig. 6 illustrates a particular mode of realization.

DETAILED DESCRIPTION

[0025] The invention will be explained in the context of a system using the USB protocol.

[0026] As illustrated in figure 1, the system comprises a USB host, which is connected to a USB device via a USB bus. The USB host communicates with the USB devices with the USB protocol.

[0027] The communication between the USB host and the USB device can be made according four different transfer modes:

- a control transfer, offering mainly a delivery and data integrity guarantee,
- an interrupt transfer, offering mainly a periodicity and data integrity guarantee,
- a bulk transfer, offering mainly a data integrity guarantee, and a possibly good data rate,
- a isochronous transfer, offering mainly a bandwidth guarantee.

[0028] Two of these four modes require a bandwidth reservation, which is accorded or not by the USB host after an enumeration phase, depending on the bandwidth already reserved by other USB devices, which are plugged onto the USB bus.

[0029] As illustrated in figure 1, the USB host comprises various applications (A0, A1). One or several services (S1, S3) are needed to run an application (A1). The application A1 needs, for example, to use the service S1 and the service S3. The services (S1, S2, S3, S4) are located on the USB device. To use a specific service S1, an application A1 uses a driver D1. The USB host also comprises a standard application (A0) associated with a standard driver (D0). Advantageously the standard application (A0) is implemented on a big number of USB hosts.

[0030] The USB device comprises various services (S1, S2, S3 and S4) in particular those (S0, S1, S3), which are needed by the applications (A0, A1) of the USB host. The USB device also comprises two applications, (A21) and (A22). Each of these applications is arranged to run the service (S2) using respectively a driver D21 and a driver D22. The application (A21) is, for example, an application arranged to be used on a first type of computer provided with a first type of Operating System. The application (A22) is, for example, the same application than (A21) but arranged to be used on a second type of computer provided with a second type of Operating System.

[0031] The USB device is, for example, a Smart Card. The USB host is, for example, a Personal Computer (PC) using a specific Operating System (OS). The services (S2, S3, S4) can be any Vendor Specific services. The service (S2) can be a service associated, for example, to an Internet Service Provider login application, (A21) or (A22). Advantageously, the service (S1) is a mass storage service. Thus driver(s) and/or application(s) contained on the USB device can be installed by simulating that the USB device is a mass storage.

[0032] As illustrated in figure 2, a service (S) can be offered, for example, at the device level, or at the interface level (S1,S2,S3,S4). The USB device also comprises a standard service (S0). Advantageously the standard service (S0) is implemented on a big number of USB devices.

[0033] As illustrated in figure 3, in a connecting step CON, the USB device is connected to a USB port of a USB host.

[0034] In a first checking step CHECK1, the USB device checks whether a negotiation flag is activated or not.

If not:

- in a first enumerating step ENUM1, the USB host will enumerate the USB device. In other words, as illustrated in figure 2, the USB host will retrieve from the USB device to the USB host only the descriptors (I) associated to the standard service S0 and to the mass storage service S1,
- in a loading step LOAD, the drivers D0 and D1 are loaded into an active memory of the USB host,
- in a negotiation step NEGO, the standard application A0 negotiates the services (S2, S3) to activate. The negotiating step comprises the following sub-steps:

- a first receiving step, in which the standard application A0 receives from the standard service S0 a first list of all the different services (S1,S2, S3, S4) which are available on the USB device,
- a comparing step, in which the standard application compares the first list of all the different services (S1, S2, S3, S4) which are available on the USB device with a second list of the services (S1,S2,S3) needed by the applications (A1,A2) to deduce the services to be activated (S1, S2, S3) on the USB device,
- a second receiving step, in which the USB host indicates which type of computer is used and which type of Operating System (OS) is used, so that the USB device can install the corresponding application(s) (resp. A21 or A22) and the associated driver(s) (resp. D21 or D22).
- a service activating step, in which the USB device activates the services to be activated, for example, by disconnecting and reconnecting the USB device to the USB host.

[0035] In an flag activating step ACTIV, the negotiation flag is activated.

[0036] In an initialization step INIT, the USB device removes its pull-up resistor in order to detach itself and then re-attach itself.

[0037] In a second checking step CHECK2, the USB device checks whether the negotiation flag is activated or not

If yes:

- in a deactivating step DEACTIV, the negotiation flag is deactivated,
- in a second enumerating step ENUM2, the USB host enumerates the USB device. As illustrated in figure 2, only the descriptors (II) associated to the services (S1, S2, S3) which have been activated and the descriptor associated to the standard service (S0) will be retrieved,
- in a second loading step LOAD2, the standard driver D0 and the drivers (D1,D21 or D22,D3) associated to the services (S1, S2, S3) which have been

activated are loaded into the active memory of the USB host.

[0038] The Smart Card (USB device) is now ready for use.

[0039] Thanks to the invention, from the user point of view, the Smart Card (USB device) is seen as a mass storage. It is then possible to install a driver (D21 or D22) and/or an application (A21 or A22) from the USB device by simulating that the USB device is a mass storage. The USB device can thus be used in any USB host, even if the driver(s) are not installed nor available, since the driver(s) is(are) available in the USB device itself.

[0040] The user can thus use the Internet Service Provider login application, (A21) or (A22), directly from the USB device to access the login service (S2).

[0041] Thanks to the invention, an Internet Service Provider can, for example, define its own proprietary login application (A21). In addition this proprietary application (A21) is stored on the Smart Card itself (USB device) and not on the computer. The risk of hacking the application (A21) is therefore reduced. The invention thus enables an enhanced security.

[0042] Note that, if a new service has to be added on the USB device, the standard service (S0) does not change and therefore the standard application (A0) does not change

[0043] As illustrated in figure 4 and 5, if the USB device is already plugged, and the user starts a new application (A3), which requires a new service (S4), which is not activated in the USB device, the standard application (A0) can negotiate the activation of the new service (S4) in a new negotiating step.

[0044] In an opening step OPEN, user opens a new application (A3) requiring a service (S4), which is not available in the current configuration of the USB device

[0045] In a negotiating step NEGO, the USB host activates the service (S4).

[0046] In a flag activating step ACTIV, the USB device activates the negotiation flag.

[0047] In an initialization step INIT, the USB device removes its pull-up resistor in order to detach itself and then re-attach.

[0048] In a checking step CHECK, the USB device checks whether the negotiation flag is activated or not.

If yes:

- in a deactivating step DEACTIV, the negotiation flag is deactivated,
- in an enumerating step ENUM, the USB host enumerates the USB device. As illustrated in figure 2, only the descriptors (III) associated to the services (S1, S2, S3, S4) which have been activated and the descriptor associated to the standard service (S0) will be retrieved,
- in a loading step LOAD, the standard driver D0 and the drivers (D1,D21 or D22,D3,D4) associated to the services (S1, S2, S3,S4) which have been activated are loaded into the active memory of the USB host.

[0049] In the above-mentioned example, the service (S4) was first hidden from the USB host point of view. But one can imagine that once the USB device is inserted in the USB host, all the services are automatically enumerated. Some drivers will not be available and the corresponding application will be marked as not correctly installed. In this case, after the enumeration step, the user will have to update the drivers from the USB device.

[0050] Figure 6 illustrates another example wherein the smartcard (USB device) comprises a loudspeaker service, a microphone service and a decryption service.

[0051] When the smartcard (USB device) is plugged into a computer (USB host) the loudspeaker service is activated by the computer. The computer first considers the smartcard (USB device) as a loudspeaker and sends an encrypted music file (1). The loudspeaker service receives the music file and sends it (2) to the decryption service for decrypting the music file. Then the decrypted music file is sent (2) to the microphone service so that the computer (USB host) believes now (3) that the smartcard (USB device) is a microphone wherein someone is speaking. The computer then sends the decrypted music file to the real loudspeaker (4).

[0052] The description hereinbefore illustrates the following features:

[0053] A system comprising a main device and an auxiliary device arranged to co-operate with each other, the auxiliary device being arranged to effect a core functionality,

the auxiliary device comprising descriptors, characterised in that the auxiliary device comprises at least one descriptor that defines a functionality that is different from the core functionality.

[0054] According to another aspect of the invention, the main device is, for example, a USB host and the auxiliary device is, for example, a USB device. It should be clear that the invention is not limited to devices communicating using the USB protocol. Other protocol like, for example, firewire based protocol may be used.

[0055] According to another aspect of the invention, the functionality that is different from the core functionality is, for example, a mass storage functionality.

[0056] According to another aspect of the invention, the functionality that is different from the core functionality is, for example, a microphone functionality.

[0057] According to another aspect of the invention, the functionality that is different from the core functionality is, for example, a loudspeaker functionality or any other functionality that is different from the core functionality, and that can be effected by the auxiliary device

[0058] According to another aspect of the invention, the auxiliary device, when it is coupled to the main device, initially presents the descriptor that defines a functionality that is different from the core functionality

[0059] According to another aspect of the invention, the USB device comprises a driver for the USB host to be installed by simulating that the USB device is a mass storage.

[0060] According to another aspect of the invention, the USB device is a smartcard or any other portable object arranged to use the USB protocol, like, for example, a Personal Digital Assistant (PDA) or a cellular phone.